In the Claims:

1. (Original) A method of encrypting information, the method comprising:

in a first pipeline stage:

obtaining a value *A* from an array having a plurality of values; and

determining a value *B* based on the value *A*;

and

in a second pipeline stage:

obtaining a value *V* from a position in the array that is based on the value *A* and the value *B*;

exclusive ORing the value *V* with a data value that forms a portion of the information.

2. (Original) The method as recited in claim 1, wherein the array is initialized using an encryption key sequence.

3. (Original) The method as recited in claim 1, wherein a first iteration of said obtaining and said exclusive ORing in the second pipeline stage is performed simultaneously with a second iteration of said obtaining and said determining in the first pipeline stage

4. (Original) The method as recited in claim 3, wherein the first iteration is based on a first value *A* in said array and wherein the second iteration is based on a next value *A* in said array.

5. (Original) The method as recited in claim 3 further comprising incrementing an index value *i* during each iteration.

6. (Original) The method as recited in claim 5 further comprising resetting the index value *i*

to zero responsive to reaching a pre-determined limit.

7. (Original) The method as recited in claim 6, wherein the pre-determined limit is 256.

8. (Original) The method as recited in claim 6 further comprising incrementing the index value *i* during each iteration prior to said resetting.

9. (Original) The method as recited in claim 1, wherein each of the plurality of values is stored in a storage location comprising flip-flops.

10. (Original) The method as recited in claim 9, further comprising shifting the array such that the value *A* is obtained from the same location in the array for each iteration.

11. (Original) The method as recited in claim 9, wherein the first pipeline stage includes a first sub-stage and a second sub-stage, wherein obtaining the value *A* is performed in the first sub-stage and said determining the value *B* is determined in the second sub-stage.

12. (Original) The method as recited in claim 9, wherein the second pipeline stage includes a third sub-stage and a fourth sub-stage, wherein an index value *g* based on the value *A* and the value *B* and the value *V* is determined in the third sub-stage, and said exclusive ORing is performed in the fourth sub-stage.

13. (Original) The method as recited in claim 9, wherein the second pipeline stage includes a third sub-stage and a fourth sub-stage, wherein an index value *g* based on the value *A* and the value *B* is determined in the third sub-stage and wherein the value *V* and said exclusive ORing is performed in the fourth sub-stage.

14. (Original) The method as recited in claim 1, wherein the array is stored in one or more register files.

15. (Original) The method as recited in claim 14, wherein the first pipeline stage includes a first substage and a second substage, wherein said obtaining the value A is performed in the first substage and said determining the value B is performed in the second substage.

16. (Original) The method as recited in claim 15 further comprising performing a swap operation in the second sub-stage, wherein the swap operation comprises switching the locations of the value *A* and the value *B*.

17. (Original) The method as recited in claim 14, wherein the first pipeline stage includes a first sub-stage and a second sub-stage, wherein said obtaining the value *A* is performed in the first sub-stage and said determining the value *B* is performed in the second sub-stage.

18. (Original) The method as recited in claim 17, wherein the second pipeline stage includes a third substage and a fourth substage, wherein an index value *g* based on the value *A* and the value *B* and a value *V* based on the value *g* is determined in the third substage, and wherein said XORing is performed in the fourth substage.

19. (Original) The method as recited in claim 1, wherein obtaining the value *B* comprises determining an index *j* based on the value *A*, wherein the value *B* is the j^{th} element of the array.

20. (Original) The method as recited in claim 19, wherein determining a value for the index *j* comprises calculating the sum of $j + A$.

21. (Original) The method as recited in claim 1, wherein obtaining the value *V* comprises determining the sum of the value *A* and the value *B* and reading the g^{th} element of the array, wherein *g* is the sum of the value *A* and the value *B*.

22. (Original) An encryption apparatus comprising:

a plurality of storage locations configured to store an array;

a first logic unit configured to read a value *A* from the array and determine a value *B* based on the value *A*; and

a second logic unit configured to read a value *V* from a position in the array that is based on the value *A* and the value *B* and to exclusive OR the value *V* with a data value that forms a portion of information that is to be encrypted;

wherein the first logic unit comprises a first pipeline stage and the second logic unit comprises a second pipeline stage.

23. (Original) The encryption apparatus as recited in claim 22 wherein the encryption apparatus is coupled to receive an encryption key sequence to initialize the array.

24. (Original) The encryption apparatus as recited in claim 22, wherein the second pipeline stage is configured to read the value *V* from the position in the array that is based on the value *A* and the value *B* and to exclusive OR the value *V* with a data value simultaneously with the first pipeline stage a second iteration of reading the value *A* from the array and calculate a value *B* based on the value *A*.

25. (Original) The encryption apparatus as recited in claim 24, wherein the first iteration is based on a first value *A* in said array and wherein the second iteration is based on a next value *A* in said array.

26. (Original) The encryption apparatus as recited in claim 24, wherein the encryption apparatus is further configured to increment an index value *i* during each iteration.

27. (Original) The encryption apparatus as recited in claim 26, wherein the encryption apparatus is configured to reset the index value *i* to zero responsive to the index value reaching a predetermined limit.

28. (Original) The encryption apparatus as recited in claim 27, wherein the predetermined limit is 256.

29. (Original) The encryption apparatus as recited in claim 27, wherein the encryption device is configured to increment the index value *i* during each iteration prior to resetting.

30. (Original) The encryption apparatus as recited in claim 22, wherein each of the plurality of storage locations includes flip-flops.

31. (Original) The encryption apparatus as recited in claim 30, wherein the encryption apparatus is configured to shift the array such that the value *A* is read from the same one of the plurality of storage locations for each iteration.

32. (Original) The encryption apparatus as recited in claim 30, wherein the first pipeline stage includes a first sub-stage and a second sub-stage, wherein obtaining the value *A* is performed in the first sub-stage and said determining the value *B* is determined in the second sub-stage.

33. (Original) The encryption apparatus as recited in claim 30, wherein the second pipeline stage includes a third sub-stage and a fourth sub-stage, wherein an index value *g* based on the value *A* and the value *B* and a value *V* based on the value *g* are determined in the third sub-stage, and wherein said exclusive ORing is performed in the fourth sub-stage.

34. (Original) The encryption apparatus as recited in claim 22, wherein each of the plurality of storage locations is a location in a register file.

35. (Original) The encryption apparatus as recited in claim 34, wherein the first pipeline stage includes a first substage and a second substage, wherein said obtaining the value A

is performed in the first substage and said determining the value B is performed in the second substage.

36. (Original) The encryption apparatus as recited in claim 35, wherein the second sub-stage is configured to perform a swap operation, wherein the swap operation comprises switching the locations of the value *A* and the value *B*.

37. (Original) The encryption apparatus as recited in claim 34, wherein the first pipeline stage includes a first sub-stage and a second sub-stage, wherein the first sub-stage is configured to obtain the value *A,* and wherein the second sub-stage is configured to determine the value *B*, and wherein the second pipeline stage includes a third substage and a fourth substage, wherein the third substage is configured to determine the value *V* and wherein the fourth substage is configure to perform said exclusive ORing..

38. (Original) The encryption apparatus as recited in claim 22, wherein the first logic unit is configured to determine an index value *j* based on the value *A*, wherein the value *B* is the j^{th} element of the array.

39. (Original) The encryption apparatus as recited in claim 38, wherein the first logic unit is configured to determine the index value *j* based on the sum of $j + A$.

40. (Original) The encryption apparatus as recited in claim 38, wherein the encryption apparatus is further configured to calculate an index value *g*, wherein the index value g is the sum of the value *A* and the value *B*, and wherein *V* is the g^{th} element of the array.

41. (Withdrawn) A method comprising:
reading a value *A* from a position in an array having a plurality of elements;
reading a value *B* from a position in the array, wherein a position in the array of the value *B* is based on the value *A*;

writing the value A into the array position from which the value B was read;

writing the value B into the array position from which the value A was read;

shifting the array such that each value stored in a position of the array is moved to another position in the array; and

repeating said reading a value A, said reading a value B, said writing the value A, writing the value B, and said shifting, for two or more iterations, wherein position in the array from where the value A is read is the same for each iteration.

42. (Withdrawn) The method as recited in claim 41 further comprising generating a value V based on the value A and the value B.

43. (Withdrawn) The method as recited in claim 41, wherein each of the plurality of elements in the array is associated with an index i, wherein the value A is read from the array position where $i = 0$ for each iteration.

44. (Withdrawn) The method as recited in claim 41, wherein each of the plurality of elements is associated with an index j, and wherein the value B is read from the j^{th} position of the array, wherein calculating the index j comprises the equation $j = j + A - 1$.

45. (Withdrawn) The method as recited in claim 44, wherein the index j is calculated by the equation $j = j + A + \mathrm{K[i]} - 1$, wherein K[i] is an i^{th} key element of a key sequence.

46. (Withdrawn) The method as recited in claim 41, wherein each of the plurality elements is associated with an index g, wherein calculating the index g comprises the equation $g = A + B - i$.

47. (Withdrawn) The method as recited in claim 41, wherein said shifting the array is performed subsequent to said writing the value A and said writing the value B.

48. (Withdrawn) The method as recited in claim 41, wherein said shifting the array is performed subsequent to said reading the value *B* and prior to said writing the value *A* and said writing the value *B*.

49. (Withdrawn) The method as recited in claim 41, wherein said shifting the array is performed subsequent to said reading the value *A* and prior to said reading the value *B*.